Moving Target Indication from a moving camera in the presence of strong parallax

Garbis Salgian, Jim Bergen, Supun Samarasekera and Rakesh Kumar
Sarnoff Corporation
Princeton, NJ 08540

Abstract

We describe an algorithm for independent motion detection from video sequences recorded from a camera moving in a 3D rich environment. Such sequences are typical in the case of Unmanned Aerial Vehicles flying at low altitude over varied terrain and also for ground vehicles. We present detection results for both scenarios.

1 Introduction

We are addressing the problem of detecting moving objects on the ground from video sequences recorded from a low-flying Unmanned Aerial Vehicle (UAV) over terrain that could have significant 3D structure. The goal is to detect independently moving objects and transmit their location (in UTM coordinates) to an Operator Control Station on the ground in real time.

Moving Target Indication (MTI) from a moving vehicle has been previously demonstrated using global parametric transformations for stabilizing the background. These techniques fail in situations in which static 3D structure in the scene displays significant parallax motion (such as video captured from a low flying UAV or from a moving ground vehicle).

For a low flying UAV, the parallax induced by the 3D structure on the ground cannot be ignored. The approach should be able to distinguish between image motion due to parallax and the one due to independently moving objects.

Large field of view (FOV) is desirable in order to be able to cover a large footprint on the ground, particularly given the low altitude of the camera. The ability to handle unrestricted camera motion is also desirable.

A practical algorithm needs to be able to handle the full range of natural environments, from planar scenes to ones with sparse 3D parallax and up to scenes with dense 3D parallax. The algorithm needs to determine the current scenario and use the model with the appropriate degrees of freedom (using an unnecessarily complex model leads to overfitting and unstable results).

Figure 1 illustrates the concept of operation for the aerial MTI scenario. On the left is the camera view with the moving targets indicated by the green bounding boxes. Once moving targets have been detected in the image, they can be projected on a map, as shown on the right of the figure. The camera footprint on the ground is marked in yellow and the location of the moving targets indicated by red circles. The transformation from camera view to map view is determined using the helicopter navigational data, camera calibration information and a terrain elevation map. This scenario can be included in a larger air-ground cooperation scenario, in which moving targets on the ground are detected from an air vehicle, their location is reported to a Command and Control center which can dispatch a ground vehicle to a suitable observation point for additional verification of the target.

We describe a real-time system that detects and tracks independently moving objects under these conditions. The system uses previously developed and new algorithms to provide detection of moving targets on the ground from a low-flying UAV. These algorithms provide robust and sensitive MTI in conditions ranging from low or zero platform motion (i.e. hovering) to rapid platform motion with rotation. They also cover the range of scene conditions from relatively flat to rough terrain with large amounts of motion parallax. The MTI application runs on a PC-104 computer on-board the CMU RMAX helicopter.

The same MTI algorithm can be used in ground vehicle applications, e.g. for situational awareness in an armored patrol in urban terrain, where the goal is to provide the user with a continuous 360 degrees view with the hatch closed, and detect moving and pop-up threads while the vehicle is moving. To illustrate this concept, we have tested the algorithm on sequences collected from a ground vehicle, and we present sample results.

The remainder of the paper is organized as follows. Section 2 discusses related work in the area of moving target detection from a moving platform. Section 3 describes the algorithm used and the implementation on the PC-104 computer. Section 4 shows results on several sequences recorded from a low-flying helicopter as well as ground-level vehicles. We conclude and discuss future work in section 5.

2 Related Work

The problem of moving target indication from a moving platform has been a very active research area in computer vision. The main challenge is to differentiate between the image motion induced by the camera moving through a static environment and that generated by independently

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 NOV 2006	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Moving Target Indication from a moving camera in the presence of strong parallax	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Sarnoff Corporation Princeton, NJ 08540**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002075., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Figure 1: Example of Moving Target Detection. **Left:** one frame from helicopter sequence; moving targets are indicated by green bounding rectangles. **Right:** Detections located on an aerial image of the same area. The camera footprint on the ground is indicated by the yellow polygon; red circles indicate the location of the two moving targets. MTI output for this frame. See text for details.

moving objects.

MTI algorithms can be broadly classified based on the method used to compensate for the camera motion into 2D and 3D algorithms. 2D algorithms assume that a global parametric transformation can align the static background over multiple frames. This is true if the camera motion is a pure rotation (e.g. for a pan-tilt camera on a fixed mount), if the scene is predominantly planar or if the amount of camera translation between consecutive frames is much smaller than the distance from camera to the scene (e.g. in the case of high altitude aerial video).

When none of the above conditions are met, the 3D structure of the scene will produce significant parallax effects that cannot be ignored.

[Irani and Anandan, 1998] present a stratification of the problem into scenarios with gradually increasing complexity for the camera induced motion: 2D scenes where a single 2D parametric transformation can stabilize the background, multi-planar scenes, with a small number of layers of parametric transformations and 3D scenes. 3D scenes are further divided into two categories: for dense parallax, use a plane plus parallax decomposition, compute the epipole, then look for points where the residual flow violates the epipolar constraint; for sparse parallax cases, the location of the epipole cannot be determined reliably, therefore the authors propose the parallax rigidity constraint as a way to determine whether two points belong to the same object (stationary background or moving object). The paper describes the individual techniques for each scenario. The parallax rigidity constraint only provides a way to determine whether two points belong to the same object (stationary background or moving object). It fails if there are no stationary objects that generate parallax (e.g. for the case of a moving object on a planar surface).

[Sawhney et al., 2000] describe an algorithm for independent motion detection for sparse 3D scenes using both view geometry and shape constraints. Their approach enforces shape constancy constraints over multiple frames.

[Ogale et al., 2005] consider three different classes of independently moving objects: *3D motion-based clustering* (similar to the epipolar constraint) for cases when the direction of the image motion of the independently moving object is different from the image motion of the stationary background induced by the camera ego-motion; *ordinal depth conflict* or occlusion-structure from motion conflict for cases when there is a stationary object closer to the camera than the independently moving object, and partially occluding it; *cardinal depth conflict* for cases when another source for determining structure is avaialable (e.g. stereo), perform cardinal comparisons between structure from motion and structure from another source.

[Jung and Sukhatme, 2004] propose a probabilistic approach for moving object detection from a mobile robot using a single camera. The ego-motion of the camera is compensated using corresponding feature sets and outlier detection, and the positions of moving objects are estimated using an adaptive particle filter and EM algorithm. An interesting aspect of this paper is that the algorithms have been implemented and tested on three different robot platforms in an outdoor environment: a robotic helicopter, Segway RMP, and Pioneer2 AT.

Other approaches for independent motion detection look for objects that have non-uniform motion. [Argyros et al., 1996]) proposes a method for the fast detection of objects that maneuver in the visual field of a monocular observer. [Nelson, 1991] presents two methods for independent motion detection. The first one is the classical epipolar constraint. The second relies on the fact

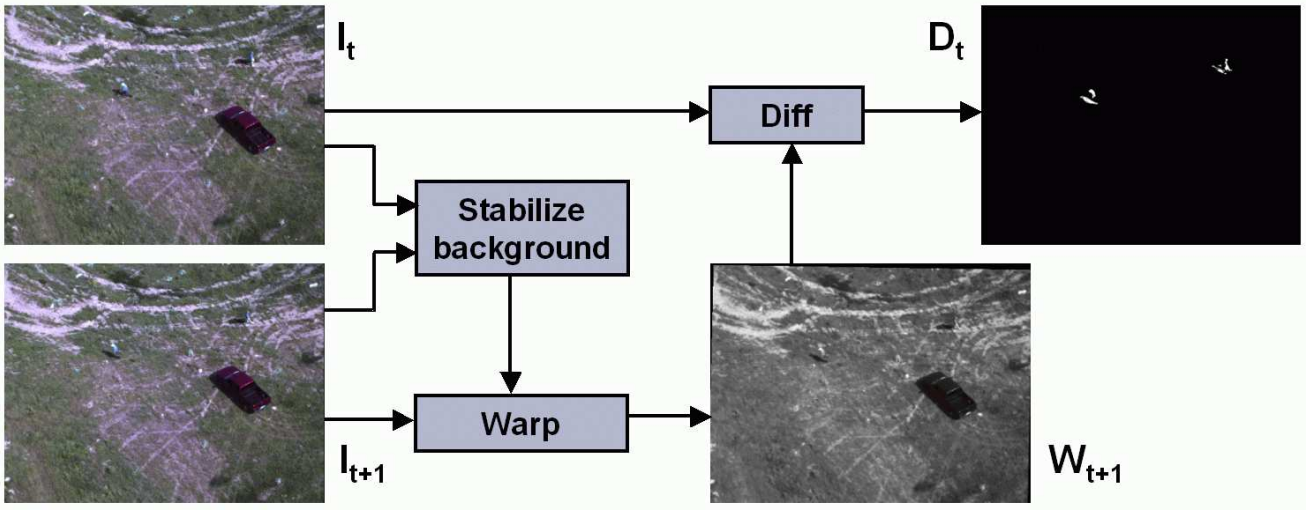

Figure 2: Global parametric alignment method for MTI.

that the apparent motion of a fixed point due to smooth observer motion changes slowly, while the apparent motion of many moving objects may change rapidly. In both cases, the qualitative nature of the constraints allows the methods to be used with inexact motion information.

If a stereo sensor is available, additional constraints can be used for detecting independent motion. In [Argyros and Orphanoudakis, 1997]), independent motion detection is formulated as robust parameter estimation applied to the visual input from a stereo camera. Depth and motion measurements are combined in a linear model whose parameters are related to the egomotion and the parameters of the stereo head. The robust estimation of this model leads to a segmentation of the scene based on 3D motion. [Talukder and Matthies, 2004] use stereo disparity fields and optical flow fields to estimate egomotion, then use predicted and observed flow and disparity to detect moving objects.

[Agrawal et al., 2005] describe a system that detects independently moving objects from a mobile platform in real time using a calibrated stereo camera. Image features are detected and tracked through the images and these tracks are used to obtain the motion of the platform. In the disparity space, two disparity images of a rigid object are related by a homography that depends on the objects euclidean rigid motion. The homography obtained from the camera motion is used to detect the independently moving objects from the stereo disparity maps.

3 Algorithm description

Detecting moving targets from a moving platform is challenging because image motion is caused by the stationary background (due to camera motion) as well as independently moving objects. An MTI algorithm needs to isolate the image motion due to the independently moving objects and ignore that due to the camera egomotion. The problem is further complicated if the scene contains significant 3D structure, since the 3D parallax can generate image motion similar to that of independently moving objects.

Below we compare two algorithms for independent motion detection from a single moving camera and discuss their relative strengths and weaknesses in section 4.

3.1 Global alignment

We use a classical algorithm [Burt et al., 1989] that looks for residual differences after aligning frames with a global parametric transformation (homography). It is well suited for detecting small moving objects on a flat surface or when camera motion is pure rotation. The method is illustrated in Figure 2 and the main steps of the algorithm are described below:

1. Register I_{t+1} with I_t, that is find a global parametric transformation M that best aligns the two images.

2. Use M to warp I_{t+1} into W_{t+1}, the frame at time t+1 compensated for the dominant motion.

3. Compute the difference image $D_t = |I_t - W_{t+1}|$. If the difference is above a threshold, label that point as belonging to an independently moving object.

Figure 3 shows sample detection results of two walking persons from about 45 meters altitude with a normal lens (Field of View around 40 degrees horizontally). In this sequence the helicopter carrying the camera was hovering, so the camera motion had no significant translational component.

When the camera is translating close to a scene with significant structure, the parallax induced by the static 3D structures in the scene represents a challenge for this algorithm. The next section discusses a method that can handle such situations.

Figure 3: Sample frame from a video sequence recorded with the CMU helicopter, showing the detection of the two moving persons.

3.2 Ego-motion plus flow orientation

This method is better suited for scenes with significant 3D structure when the camera motion has a significant translational component. The first step is to recover the camera motion in 3D. Next, eliminate the image motion due to the camera rotation (which is independent of the 3D structure in front of the camera) and compute the residual optical flow. This flow will be epipolar, i.e. all flow vectors corresponding to the static background will intersect at a common point (the epipole). The points in the image where the flow vectors do not satisfy this constraint are labeled as independently moving.

The input consists of two frames from a calibrated video sequence. The time separation between input frames could be adjusted depending on the speed of the camera through the environment, but for this discussion we will assume that input frames are consecutive: I_t and I_{t+1}.

Camera calibration information is encapsulated in the intrinsic parameter matrix K:

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix},$$

where f_x, f_y are the camera focal length in the horizontal and vertical direction (in pixels) and c_x, c_y are the image coordinates of the camera center.

The 3D camera motion between I_t and I_{t+1} is estimated using a visual odometry algorithm [Nister et al., 2006]. For more robust visual odometry results, multiple cameras with fixed relative geometry may be used. This increases the total Field Of View of the system and the probability that reliable features can be detected and tracked as the cameras move through the environment. The output of the visual odometry algorithm consists of the camera 3D rotation and translation estimates (R and T) over time.

Given two input frames I_t and I_{t+1} and R, T that describe the camera motion between time t and $t+1$, the main steps of the algorithm are described below:

1. Eliminate the image motion component due to camera rotation. This motion is independent of the 3D structure of the scene, and can be computed as a global parametric transformation $M = KRK^{-1}$, where K is the camera matrix and R is the 3D rotation. Use M to warp I_{t+1} into W_{t+1}, the frame at time t+1 compensated for camera rotation.

2. Compute optical flow between I_t and W_{t+1}. Since the effects of the camera rotation have been eliminated, this flow will be epipolar, i.e. all flow vectors corresponding to the static background will intersect at a common point (the epipole). We use the coarse-to-fine approach for optical flow computation described in [Bergen et al., 1992].

3. Compute the epipole location from the translation component of the 3D camera motion and camera matrix: $e = KT$

4. For every point where an optical flow vector has been computed, compare its orientation with the epipolar direction. If the difference is above a threshold, label that point as belonging to an independently moving object.

Figure 4 illustrates the main algorithm steps on a frame from a sequence recorded from a helicopter. The camera was pointing forward and pitched down 45 degrees, and the helicopter was moving forward. The scene is mostly static, except for two moving persons. The flow between I_t and W_{t+1} (step 2) is shown in the upper right part (only a subset of the flow vectors is displayed for clarity). The flow vectors that agree with the epipolar direction are colored green, while the ones that don't are colored red. These vectors are assumed to correspond to independently moving objects. The instantaneous detection map D_t is generated based on the magnitude of the angular difference between the predicted (epipolar) and actual direction of the flow vectors.

3.3 System Implementation

The two algorithms have been implemented in C/C++, and the code can be build on a general-purpose PC under Windows or Linux Operating System. The optical flow computation uses the Intel Performance Primitives library for acceleration.

The computing platform for the aerial examples is a Pentium-M 1.8GHz PC-104 computer, with a firewire card for capturing video from a digital camera. Since the helicopter platform exhibits high vibration during flight, only solid-state storage is reliable. A 4GB Flash Card holds the Linux OS and the application, leaving about 2GB for recording video for test purposes. The application captures images from the firewire cameras and receives helicopter

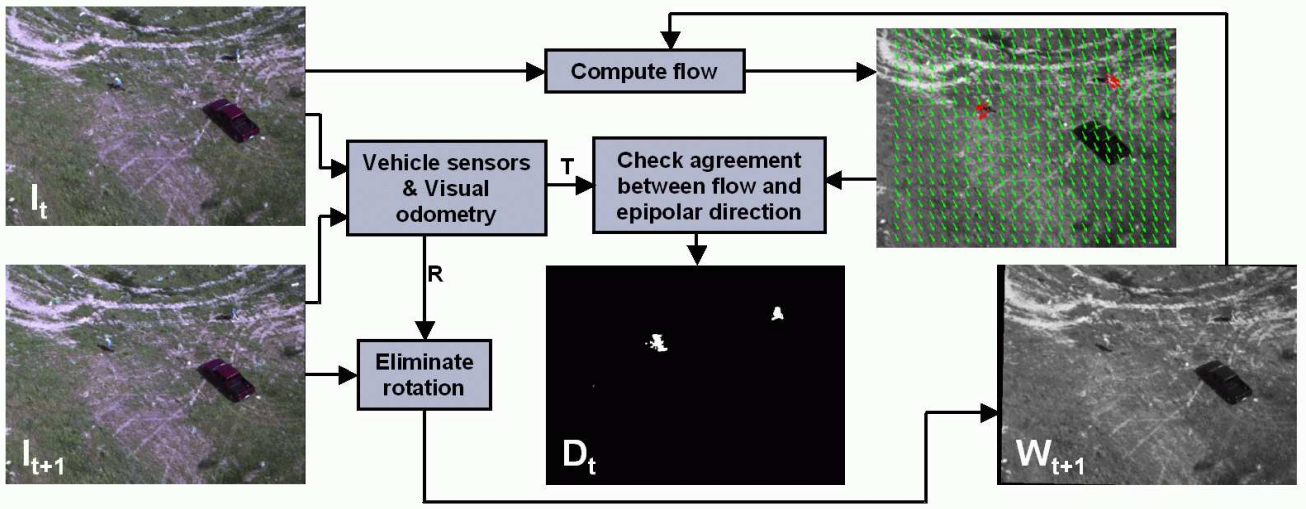

Figure 4: 3D MTI method.

pose information over Ethernet from the navigation computer. In data collection mode, it stores the video and metadata on the FlashCard. In live operation mode, the MTI algorithm is executed and the detection results together with a reduced resolution version of the input video are sent over wireless Ethernet to a ground station for display.

For the ground vehicle examples, we used a general-purpose PC to collect the video and vehicle metadata and processed the sequence off-line.

4 Examples

In this section we present several examples of the output obtained from the MTI algotrithm on sequences collected in different scenarios.

4.1 Air vehicle scenario

We tested the MTI algorithm on multiple monocular video sequences collected using the CMU helicopter at Ft Indiantown Gap. The scenarios recorded so far include people and vehicles moving in open areas, next to tree lines or along roads in wooded areas, with the helicopter flying at about 50m above the ground.

Figure 5 compares the output of the two methods. In this sequence the helicopter is flying over a wooded area along a dirt road. There are two moving objects in the scene, a HMMWV and a person.

We show a few representative frames, and sample output on each row in the figure. On the left is the input frame, and two the right are the corresponding instantaneous detection maps obtained with the two methods: the 3D MTI method in the center and the global alignment method to the right. Black corresponds to no detection, and bright areas correspond to moving objects.

For the first frame (top row in Figure 5), the scene is a flat field with no moving objects, and both methods produce the correct output (the solid black detection map indicates no moving objects). The frame on the second row contains a moving person which is detected by both algorithms. Note that the global alignment method produces a sharper definition of the moving object than the 3D MTI method, due to the local support window used in the optical flow computation.

In the third row, the scene has no moving objects, but there is significant 3D structure (tall trees). The 3D MTI algorithm is not sensitive to the static structure, while the global alignment method generates false alarms (white regions in the detection map corresponding to the tall trees). Finally, the fourth row shows an example of 3D static structure and moving objects. In the output for the global alignment method (rightmost) the moving person and vehicle are well defined, but there is also a significant response on the tall trees, due to the parallax generated by the 3D structure of the scene. For the 3D MTI method (center), only the person and vehicle are detected, there are no false alarms on the trees.

Figure 6 shows sample output for two additional sequences. For each sequence, a representative frame is presented on the left, with the yellow arrows pointing to interesting objects in the scene. The center image is the processed frame with the bounding box for the tracked objects overlaid. The image on the right is the MTI output; brighter points correspond to regions with higher likelihood of belonging to independently moving objects. The moving person and moving vehicle are detected in both examples (indicated by the overlaid bounding box), while the tall trees do not generate false alarms.

4.2 Ground vehicle scenario

We have also investigated applying the same 3D MTI algorithm to sequences collected from a moving ground vehicle. The ground vehicle case is more challenging than the low-altitude aerial one since the distance to the closest point in the 3D scene is smaller and therefore the parallax

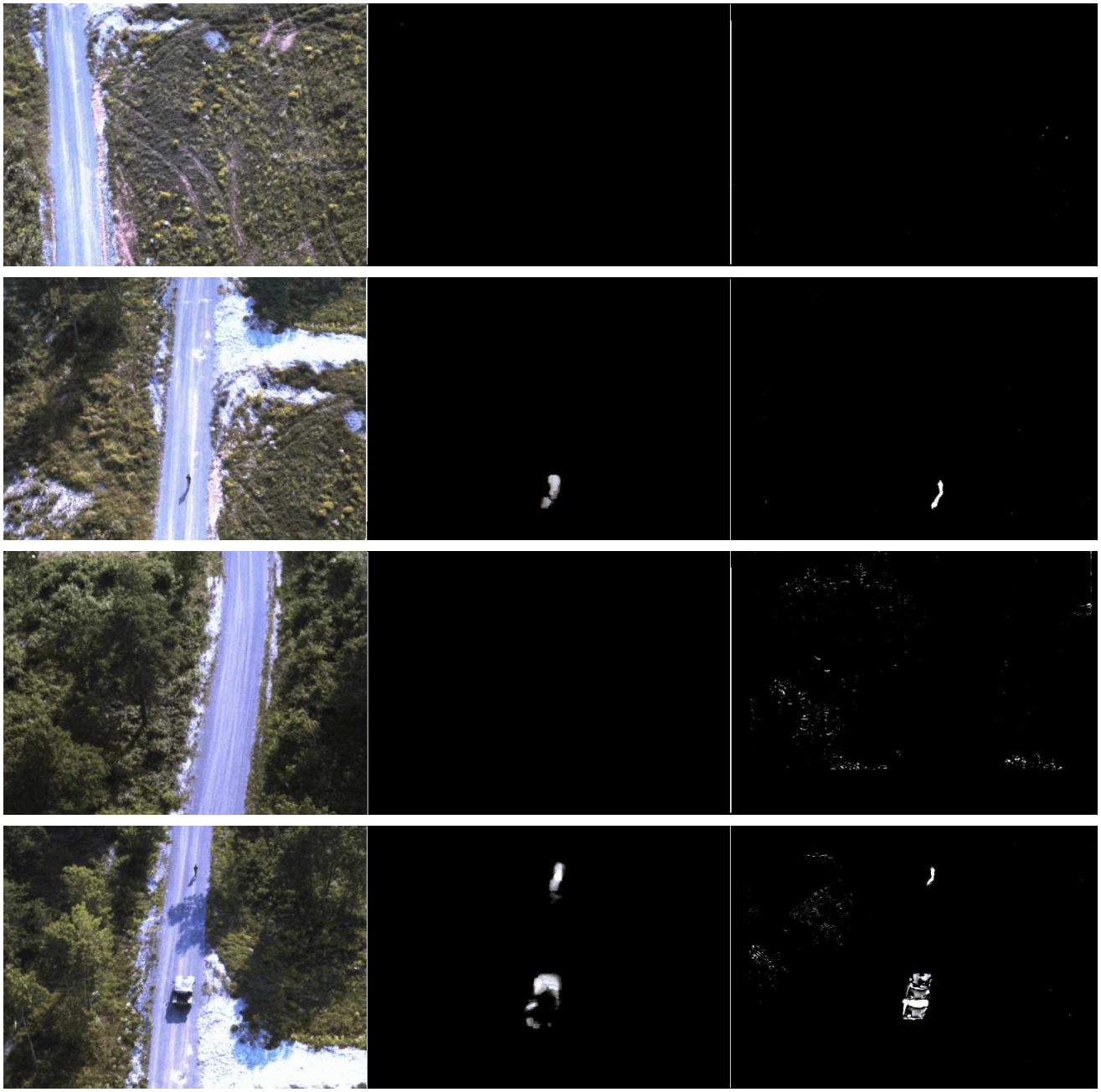

Figure 5: Left: One frame from a helicopter sequence. **Center:** 3D MTI method **Right:** Global parametric alignment method. See text for details.

Figure 6: Examples from two helicopter sequences (top row tree line sequence, bottom row road sequence). **Left:** one frame from helicopter sequence; **Center:** Processed frame with bounded boxes of tracked objects overlaid. **Right:** MTI output for this frame. See text for details.

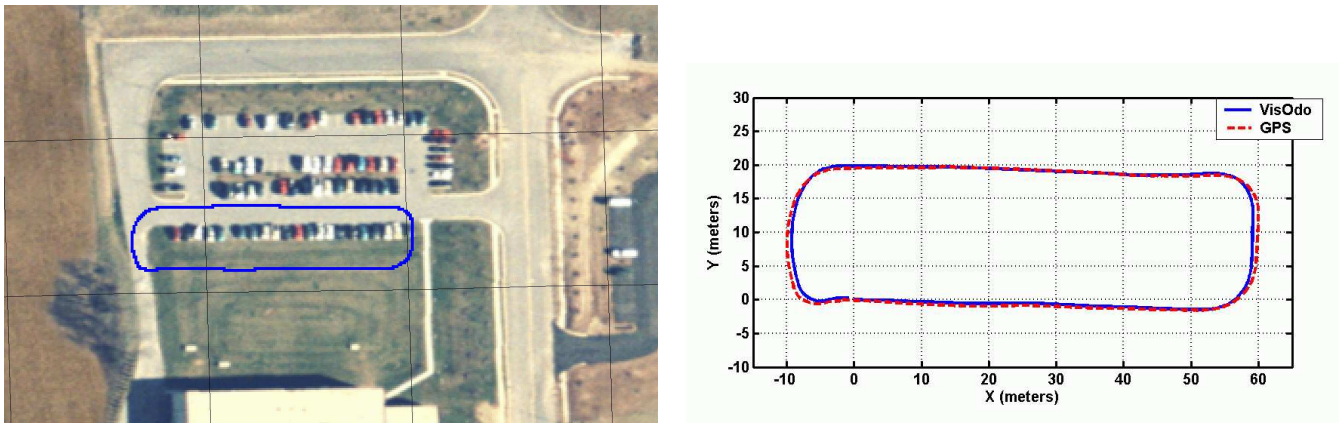

Figure 7: XUV trajectory for the ground level MTI example.

Figure 8: Sample detection results for ground level MTI. The red boxes denote image areas where independent motion was detected.

effects are stronger.

Data was collected with a stereo head mounted on the pan-tilt unit of an XUV, in the GDRS parking lot in February 2006. As the XUV drove along the trajectory shown on the left side in Figure 7, several people moved in front of the vehicle at distances ranging from 5 to 30 meters and with speed from walking and up to running.

Camera motion over time was recovered using visual odometry on both cameras in the stereo pair for increased robustness. The right side of Figure 7 shows the trajectory obtained from the vehicle INS system compared to the trajectory recovered from visual odometry. Next, the 3D MTI algorithm was applied to the left camera only. Figure 8 shows sample detections on a few frames from the sequence. The red overlay boxes indicate regions where the image motion was determined to be inconsistent with the camera motion through a static environment, and therefore are labeled as independently moving objects.

5 Conclusion

We presented a system for independent motion detection from a moving platform in the presence of strong parallax and showed examples from two scenarios with strong parallax: low-flying air vehicle and ground vehicle. The system can use multiple MTI algorithms each best suited for different operating conditions.

Since most of the test environments encountered so far contain significant 3D structure which generates strong parallax effects, we have been using mostly the 3D MTI algorithm described in section 3.2. Future work will address the problem of automatic switching between MTI algorithms based on terrain conditions and camera motion.

Acknowledgements

The work described in this paper was performed under the Robotic Collaborative Technology Alliance program sponsored by the Army Research Laboratory. The authors would like to thank the following Alliance partners: CMU autonomous helicopter team lead by Dr. Omead Amidi for providing all the aerial sequences presented in this paper and for integrating the aerial MTI system on the RMAX helicopter for a live demonstration; General Dynamics Robotics Systems for providing an XUV and logistical support for collecting the data used in the ground vehicle example.

References

[Agrawal et al., 2005] Agrawal, M., Konolige, K., and Iocchi, L. (2005). Real-time detection of independent motion using stereo. In *IEEE workshop on Motion*, Breckenridge, Colorado.

[Argyros et al., 1996] Argyros, A. A., Lourakis, M. I. A., Trahanias, P. E., and Orphanoudakis, S. C. (1996). Fast visual detection of changes in 3d motion. In *IAPR Workshop on Machine Vision Applications*, Tokyo, Japan.

[Argyros and Orphanoudakis, 1997] Argyros, A. A. and Orphanoudakis, S. C. (1997). Independent 3d motion detection based on depth elimination in normal flow fields. In *Computer Vision and Pattern Recognition*, pages 672–677, San Juan, PR.

[Bergen et al., 1992] Bergen, J. R., Anandan, P., Hanna, K. J., and Hingorani, R. (1992). Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, Santa Margharita Ligure, Italy.

[Burt et al., 1989] Burt, P., Bergen, J., Hingorani, R., Kolczynski, R., Lee, W., Leung, A., Lubin, J., and Shvayster, H. (1989). Object tracking with a moving camera. In *Workshop on Visual Motion*, pages 2–12, Irvine, CA, USA.

[Irani and Anandan, 1998] Irani, M. and Anandan, P. (1998). A unified approach to moving object detection in 2d and 3d scenes. *IEEE Transactions onPattern Analysis and Machine Intelligence*, 20(6):577–589.

[Jung and Sukhatme, 2004] Jung, B. and Sukhatme, G. S. (2004). Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *Intelligent Autonomous Systems*, pages 980–987.

[Nelson, 1991] Nelson, R. C. (1991). Qualitative detection of motion by a moving observer. *International Journal of Computer Vision*, 7(1):33–46.

[Nister et al., 2006] Nister, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applicatio. *Journal of Field Robotics*, 23(1).

[Ogale et al., 2005] Ogale, A. S., Fermüller, C., and Aloimonos, Y. (2005). *Detecting independent 3D movement*, volume Handbook of Geometric Computing of *Applications in Pattern Recognition, Computer Vision, Neural computing, and Robotics*. Springer Verlag.

[Sawhney et al., 2000] Sawhney, H., Guo, Y., and Kumar, R. (2000). Independent motion detection in 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1191–1199.

[Talukder and Matthies, 2004] Talukder, A. and Matthies, L. (2004). Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. In *Conference on Intelligent Robots and Systems*, Sendai, Japan. IEEE.